

R behind the scenes: Using S the (un)usual way

Friedrich Leisch

Institut für Statistik
Ludwig-Maximilians-Universität München

ISM, Tokyo, 7.12.2007



Genesis & Acknowledgements

- Most of what I am talking about here has been implemented by others.
- More than two years ago Sigbert Klinke tried to put together an invited session on statistical software for the ISI session in Lisbon 2007. I was supposed to talk about dynamic statistical documents.
- Then the ISI executives began to re-arrange proposals, merged our session proposal with one on “algorithms”, and cut down the number of speakers step by step.
- In the beginning we tried to match that by including cut-out topics into the remaining talks, in the end I could have talked about anything at all (but the abstract was fixed by then).
- However, others seemed to like the talk (or at least the abstract), so here I am ...

Overview

- Using R as a scripting language and text preprocessor
- Embedding R in other applications
- Talking to R
- Alternative R frontends: web pages, spreadsheets, user interfaces
- Legalese

Friedrich Leisch, 7.12.2007, R behind the scenes

1

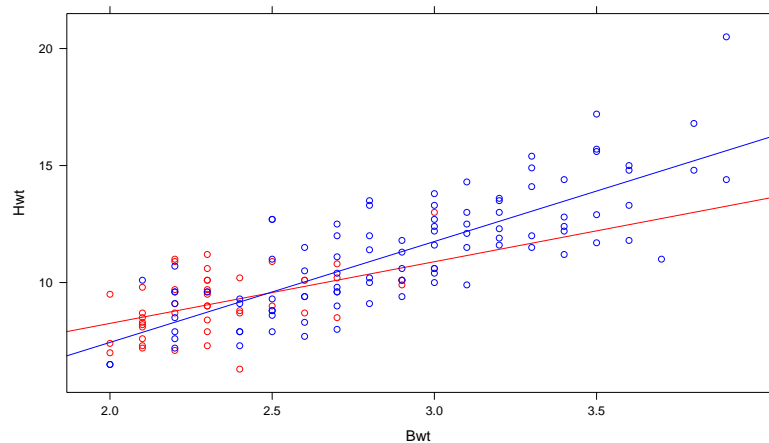
Common R Usage

```
> data(cats, package = "MASS")
> lm1 = lm(Hwt ~ Bwt * Sex, data = cats)
```

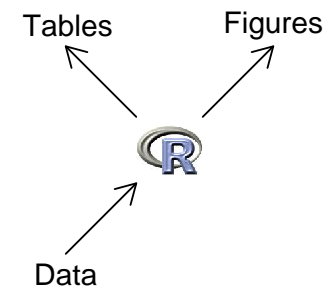
results in:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.9813	1.8428	1.62	0.1080
Bwt	2.6364	0.7759	3.40	0.0009
SexM	-4.1654	2.0618	-2.02	0.0453
Bwt:SexM	1.6763	0.8373	2.00	0.0472

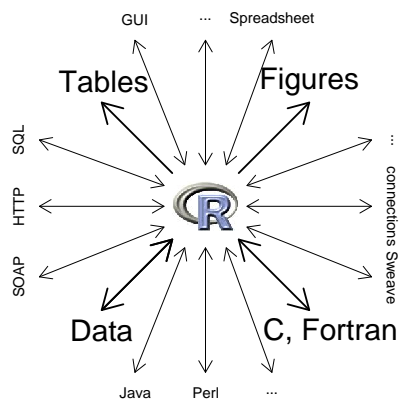
Common R Usage



Common R Usage



Common R Usage



S Programming

- R is an S interpreter, and S is a full-featured programming language.
- E.g., we have built a lot of string processing and graphical capabilities into R, hence you can use it for a lot of things that have nothing to do with statistics:
 - create a web gallery of your vacation photos
 - write serial emails
 - write presentation slides
 - check R code against documentation
 - create a poster of “Red Mount Fuji”
 - ...

Sweave

- Embed complete R code producing numbers and figures into your text document (L^AT_EX, HTML, OpenOffice).
- Used as a tool for reproducible research, to write manuals that can be checked (package vignettes), presentation slides, ...
- Part of every R installation (package `utils`).
- Sweave is plain R, runs on all platforms.

Sweave: New(er) Features

- BioC package `weaver` (by Seth Falcon) allows caching of computation results.
- CRAN package `patchdvi` can backtrack location from DVI viewer to Sweave source file, Sweave in R starting with 2.5 series can keep original formatting of code and noweb chunk references (all by Duncan Murdoch).
- We are in the process of starting a separate (but recommended, hence omnipresent) package `Sweave`, integrating the above, manuals and FAQ from my homepage and more into a single location.

Sweave: New(er) Features

I am currently playing around with evaluation of code chunk options:

```
<<>>=  
fig.pdf <- TRUE  
fig.width <- 2*pi + sin(1)  
eval.on.friday <- ( length(grep("^Fri", date())) > 0 )  
@  
  
<<fig=TRUE, width=fig.width, pdf=fig.pdf, eps=!pdf>>=  
...  
@  
  
<<eval=eval.on.friday>>=  
...  
@
```

Embedding R: Shared Library

- From an authors point of view Sweave “embeds” R code into L^AT_EX.
- But as R is still the master processor, this is not really “embedding” in a computer science sense.
- For some time now, it is surprisingly simple to really embed R in other applications (thanks to Duncan Temple Lang, Brian Ripley, Simon Urbanek, ...).
- As R itself is written in C, you can use R as part of your own C/C++ programs.

Embedding R: Sockets

- If you want to embed R into applications written in other programming languages than C, interfaces become more of an issue.
- There are several binding to other languages like Perl or Java.
- Another approach, which is more language-independent is to “talk” to R via a socket connection or other communication protocols (DCOM, SOAP, ...).
- Again, this is much simpler than most users think.
- Package Rserve implements a full featured server including data exchange.

Applications

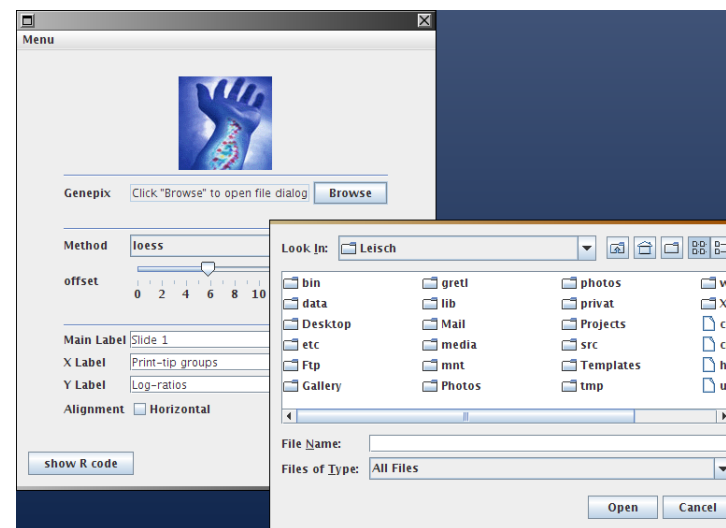
- Spreadsheets: RExcel (T. Baier & E. Neuwirth)
- Alternative GUIs: ESS, R Commander (J. Fox), JGR (Uni Augsburg), SciViews (P. Grosjean), Statistiklabor (FU Berlin), ...
- Dynamic web pages: Rpad (Tom Short, PG), MediaWiki (Sigbert Klinke), ...

RGG: R GUI Generator

Master Thesis of I. Visne (ARC Seibersdorf): Specify GUI using mixture of XML tags and R commands, automatically generate Java GUI components:

```
<rgg>
  
  <separator title="Input" span="3"/>
  targets = readTargets(
  <filechooser label="Genepix" description="Text Files (*.txt)"
    extensions="txt" fileselection-mode="files-only"/>)
  RG = read.maimages(targets$FileName, source="genepix")
  ...
</rgg>
```

RGG: R GUI Generator



Legalese

- R is free software and ships under the terms of the FSF GPL.
- The GPL is as much a political statement as a license for software distribution.
- The GPL is an infectious disease: Applications that are “derived work from” or “linking to” GPL software must also be distributed under the GPL (or compatible licenses).
- I am not a lawyer, don’t ask me (or other R core members) on details!
- The only advice: If you don’t distribute your software, you don’t need to think about licenses. But please do distribute and give back to the community!

Summary

- There are numerous other ways of using R than through the standard prompt.
- R is much more than “software for data analysis”, it can be the software equivalent of a Swiss army knife like many other scripting languages.
- The R community is very active in exploring the possibilities, and we certainly have only seen the tip of the iceberg yet.
- If you think that using R the way it is is not optimal: do improve on it (R core most likely won’t scratch your itch)!!!

Paper: Proceedings of the 56th Session of the ISI, 2007
(or drop me an email at Friedrich.Leisch@lmu.de).